Spring 5-1-2020

# Temporal Localization of Video Topics Using the YT8M Dataset: An Exploration

Katherine Riedling
katherine.riedling@uconn.edu

UCONN | SCHOOL OF ENGINEERING

# Temporal Localization of Video Topics Using the YT8M Dataset: An Exploration

Katherine Riedling
Computer Science & Engineering
University of Connecticut
Honors Scholar Thesis, May 2020

Thesis supervisor: Joseph “Joe” Johnson
Honors advisor: Zhijie Jerry Shi

# Temporal Localization of Video Topics Using the YT8M Dataset: An Exploration

Katherine Riedling
katherine.riedling@uconn.edu
University of Connecticut
Storrs, Connecticut

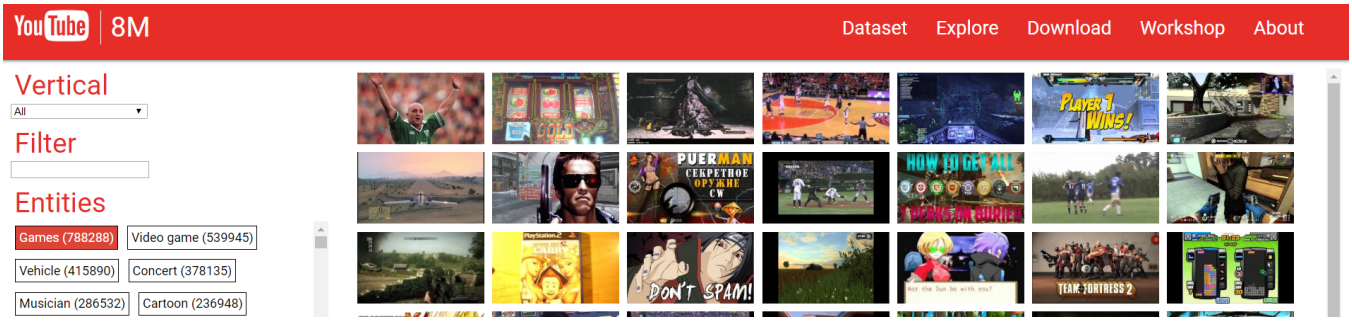


**Figure 1: Screenshot of the "Explore" tab on the YT8M site.**

## ABSTRACT

Due to the progress made in computing resources and artificial intelligence, applications in computer vision have gained a lot of traction over the past decade. One such application applies to video understanding and content analysis, which are the main goals of the annual YouTube-8M Video Understanding Challenge. In the newest challenge, the aim is to localize events to specific video segments in addition to discerning the main topics of the video. This paper introduces and presents a broad overview of techniques, data, and the top-performing algorithms presented at the International Conference on Computer Vision (ICCV) conference last year. Ensemble methods and candidate generation with VLAD model representation appear to be particularly popular and high-performing. There are many extensions of this line of research, including further time and budget constraints, as well as content-specific tasks. Implications extend beyond user content searches to optimizing content retrieval and analysis.

## CCS CONCEPTS

• **Computing methodologies** → **Classification and regression trees**; **Neural networks**; **Video summarization**; **Visual content-based indexing and retrieval**; **Activity recognition and understanding**.



, ,



ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

## KEYWORDS

computer vision, video understanding, classification, neural networks, deep learning

## 1 INTRODUCTION

Computer vision has a vast array of applications, from gaming to biomedical services. Object recognition can already address, for example, product defect detection or license plate recognition.[2] Action localization for a video, however, could extend these applications to respectively catch the moment a product breaks down or the exact time at which a driver loses control of a vehicle. One application domain that generates a lot of interest is video streaming, which, among others, includes the goal of "recommending new videos or automatic video classification" [19]. It also entails "improved video search (including search within video), video summarization and highlight extraction, action moment detection, [and] improved video content safety." [20] For example, videos subject to review could be more easily tracked for violations of community guidelines. Users could find information much more quickly, and content creators could find editing and other post-processing mistakes. Content analysis of newsreels and security footage could be done much more efficiently. There are also positive implications for accessibility features, such as content summaries and timestamps in addition to closed captioning. There are, however, also ethical concerns in certain application domains like data privacy. For example, surveillance policies from CCTV footage to work productivity managers can exploit action localization to incriminate their samples' constituents, but such considerations are beyond the scope of this project.

### 1.1 Problem and Scope

The YouTube-8M (YT8M) Video Understanding Challenge addresses label classification for YouTube videos through the prediction of

video-level topic annotations. In its first year, participants were tasked with building the models that would predict these topic labels.[18] In its second year, the challenge aimed to build the same models but under budget constraints, namely a size limitation of 1GB for models. [19] This paper will focus, however, on the 3rd YouTube-8M Video Understanding Challenge, which addresses temporal concept localization.[20] In other words, video topic labels are not only predicted on a video level, but also on a segment level. One important distinction between all of the tasks and object recognition is the scope; in object recognition, all important items are identified. In video understanding, the key idea is that important objects are identified and described with video labels in order to summarize the video content succinctly and effectively.[3] For instance, suppose a dog owner uploads a video of their dog playing with a ball in a park. If every item were to be identified in a given frame, there would be a lot of information about contextual items, such as the fountain, pedestrians, trees, etc. For video understanding, only the dog and the ball would be relevant to identifying the main theme of the video.

## 1.2 Dataset

The YT8M benchmark dataset was created with the goal of "removing computational barriers by pre-processing the dataset and providing state-of-the-art frame-level features to build from" and is the largest "multi-label video classification dataset" as of 2016. [3] In its original form, it consisted of over 8 million videos represented by 4800 Knowledge Graph entities (a significant improvement over the 500 in other datasets), with "pre-computed state-of-the-art features for 1.9 billion video frames." [3] These computational additions were intended to help advance computer vision research with respect to video content analysis, akin to a video analog for ImageNet.[18] ImageNet assigns word phrases called "synonym sets" ("synsets") to different topics, which are represented by different images.[9] Similarly, the YT8M dataset assigns annotations to different videos. YouTube data were used for the YT8M set for their diversity and abundance, though they are also noisy. [3]
The dataset was constructed in the following pipeline:[3]

(1) First, the vocabulary was built using a Knowledge Graph containing millions of topics under multiple category types to describe the "main themes" of a video. This graph was reduced based on whether labels could be determined by looking at the content of each video, and was subsequently curated by human observers, who rated the difficulty of identifying objects based on the visual information presented in-frame alone.
(2) The YouTube annotations were used as a retainment criterion to randomly sample 10 million videos with at least 1000 views and that were between 120 and 500 seconds long. Another criterion was the amount of available training data, so the set was further pruned until each topic had at least 200 video examples. Data were split in a 70-20-10 ratio for training, validation, and testing respectively.
(3) Videos were pre-processed by applying the Inception network for feature extraction, pre-trained on ImageNet, to decode six minutes per video at one frame-per-second using ReLu activation layers, followed by PCA and whitening. Frame- and video-level labels were compressed. Training and validation data are made public, though test data were withheld for the competition.

The YT8M Segments dataset used in the third and most recent challenge is an extension of the original set, including not only frame- and video-level data, but also segment-level data for 6.1 million public, "popular", uniformly sampled videos with an average of 5.0 segments per video.[21] In other words, 237,000 segments spanning 1000 classes have been manually curated in order to allow for segment-level temporal localization. Evaluations are performed on the test data of the YT8M dataset and ranked using the mean average precision (mAP) @K=100,000 metric, rather than using global average precision (GAP).[20][19][18] The vocabulary, a subset of the vocabulary base used in 2018's challenge, was adjusted to filter out videos that cannot be localized, such as annotations that span for the entire duration of a video. The most current version of the dataset contains 3862 classes (Knowledge Graph entities), 2.6 billion video-, frame-, and segment-level audiovisual features, with 1.3 billion audio features and 1.3 billion visual features. Visual features, temporally localized at the frame level, were extracted through the Inception-V3 model. Audio features were extracted using an acoustic model from Hershey et al. [7] The least popular entity, the Cylinder topic, has only 123 examples. The second least popular entity, Mortar, has 127 examples. All entities are distributed across 24 verticals or categories (Fig. 2). [21]

## 1.3 Concept Overview

Domains within computer vision can also be quite broad. Since many important concepts are common to multiple methods discussed in this literature review, they will be summarized in this section for the sake of clarity and brevity.

- Action classification and localization - As the term "action" implies, it involves classifying an event in a video and determining when it took place.

- PCA/whitening: Principal component analysis (PCA) is a dimensionality reduction technique that pares down high dimensional data into its principal component vectors, i.e. into a lower-dimensional space.

- Fisher vector (FV) - The Fisher vector is a normalized gradient vector of the Fisher score of one or more feature vectors, where the gradient indicates the direction in which the parameters should move.[17] The FV encoding can be incorporated into neural networks called FVNets.[16]

- VLAD - Vector of Locally Aggregated Descriptors (VLAD) is an image representation model that aggregates "local features into a vector of fixed dimension," where feature vectors - through $k$-means training - are assigned to the closest representative vector (known as a visual word, VW).[17] The feature vectors are then found and extracted, before being quantized into VWs. [17] NetVLAD is a portmanteau of its

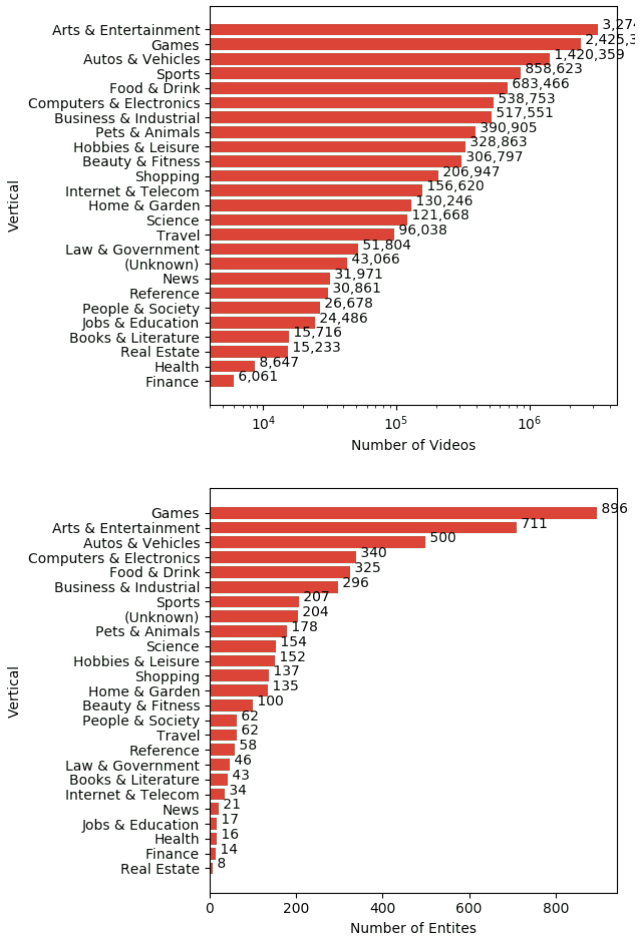


**Figure 2: Distributions of samples (top) and topics (bottom) over the verticals, i.e. categories. Images from Google Research. (https://research.google.com/youtube8m/index.html).**

constituents, a neural network with the image representation VLAD.[10] [**?** ]

- Convolutional neural networks (CNN) - CNNs are often applied to grids, as of pixels in images. [5] Each layer in a CNN has three important stages, the first of which is convolution - an operation that takes in some data, applies a kernel (typically a probability density function), and returns a feature map. The second "detector" stage takes the linear activation result (feature map) and applies a nonlinear activation function to it.[5] RELU activation functions are particularly popular NN activation functions.[4] The last stage is pooling, where the layer output is modified. In max pooling, the maximum output in a certain subgrid is output.[5]

- Recurrent neural network (RNN) - RNNs are particularly adept at processing sequences. It performs convolution in the first dimension and uses the past sequence states as input to get to the current state, which is known as "unfolding."[5] One important note is that training can be parallelized, and it can map input to output sequences of different lengths.[5]

- Two-stream architecture - Two-stream CNNs are extensions of deep CNNs that separate video from audio streams, where CNNs are applied to both streams such that the "spatial stream performs action recognition from still video frames, whilst the temporal stream is trained to recognize action from motion in the form of dense optical flow." [15]

- Online learning - Online learning, which includes stochastic gradient descent, refers to learning methods that process data incrementally either due to the nature of the application or to the volume of data (which exceeds the memory capacity).[14]

- Knowledge distillation - Knowledge distillation refers to the transfer of knowledge between a "teacher" network and its "student" network, which is a simpler model and thus allows for model compression. [11][16]
- Mixture-of-Experts (MoE) classifier - A MoE classifier uses a network of gates for "soft" switching between different "expert" networks, in which learning requires both the parameters of each expert and the parameters of the gating network to be learned.[6]

- Mean Average Precision (MAP) - The Mean Average Precision is the average precision taken over all predictions in all videos. It represented by the following equation:

$$\sum_{c=1}^{C} \frac{\sum_{k=1}^{n} P(k) * rel(k)}{N_c} \tag{1}$$

Where "$C$ is the number of Classes, $P(k)$ is the precision at cutoff $k$, $n$ is the number of segments/class, $rel(k)$ is an indicator function equaling 1 if the item at rank $k$ is a relevant (correct) Class, or zero otherwise, and $N_c$ is the number of positively-labeled segments" for each class.[20]

- Global Average Precision (GAP) - The global average precision is the average precision taken over all predictions in all videos. It represented by the following equation:

$$\sum_{i=1}^{N} p(i)\Delta r(i) \tag{2}$$

Where $p(i)$ and $r(i)$ denote the precision and recall respectively, N is "the number of final predictions", and $k$ denotes the number of predictions per video (the "highest k confidence scores per video").[19]

## 2 LITERATURE REVIEW

Although the scope of 2019's competition differs from that for the two years before it, some of the approaches implemented impacted subsequent research. Those will be discussed here, as well.

## 2.1 Summary

The winning approach in 2017's competition used an ensemble of "learnable pooling techniques" like soft bag-of-words, Fisher vectors, NetVLAD, GRU, and LSTM for aggregation, as well as a learnable nonlinear network unit for context gating.[13] The pipeline works as follows: Input features are extracted, with RELU activation functions and the last layer fully-connected in the Inception network for visual features. Audio features are extracted using a CNN. PCA and whitening are applied to the result. Next, a two-stream architecture handles pooling into a 1024-dimensional representation. Lastly, a soft MoE classifier is applied to take videos as input and output corresponding label sets and scores. The classifier is followed by a context gating layer that reweights output class probabilities based on the training results.[13] Learnable pooling models were structured to simplify NetRVLAD because there are fewer parameters. Pooling was "via clustering," using soft descriptor assignments for bag-of-visual-words, VLAD, and Fisher vector representations.[13] The context gating layer leverages a Gated Linear Unit (GLU) to transform feature vectors. Training involves the ADAM Algorithm with a cross-entropy loss function (Fig. 3).[13]

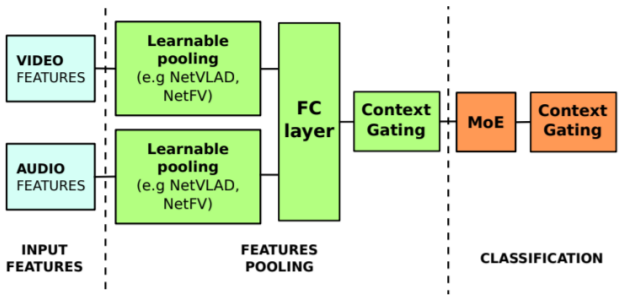


**Figure 3: The solution architecture for 2017's YT8M 1st place winner, taken from its corresponding conference paper.[13]. "FC" denotes a fully connected layer.**

The winning approach in 2018's competition, like high-performing solutions prior to it, used ensemble learning. However, model distillation and quantization were applied in order to reduce the size of the model. The authors note that larger models, ones with bigger clusters and hidden layers and exceed the 1GB size limitation, tendentially perform better. [16] The component models apply NetVLAD, Fisher vectors, bagging (Deep Bag-of-Frames), and recurrent neural network families.[16] They "sampled 300 frames with replacement during training" and used exponential decay averaging for the stored checkpoint weights. They first applied model distillation to compress the model after training, which was done with a larger teacher network. And because of the tradeoff in performance with model quantization, they used partial weights 8-bit quantization on variables with over 17,000 elements only (such as the weights for fully connected layers).[16] Submodels were separated by family, and the best three models per family were ensembled by equal weighting. Predictions were "used to generate soft targets for a distillation dataset," and then the model architectures were re-trained on the distillation datasets (Fig. 4).[16]

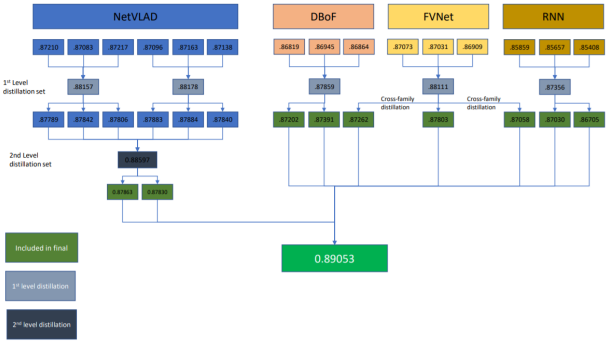


**Figure 4: The solution architecture for 2018's YT8M 1st place winner, taken from its corresponding conference paper.[16]. "DBoF" refers to the Deep Bag-of-Frames approach.**

The third-place approach for that year was the basis for its authors' approach in 2019's competition (which also placed third). Their paper introduces NeXtVLAD, an improvement for NetVLAD because it converges faster and is more robust to overfitting.[10] The NeXtVLAD algorithm accepts frame-level features as input and decomposes them into a lower dimension. It encodes and aggregates the result. The approach also builds upon the winning approach of 2017 by applying a gating module (SE Context Gating module) for "modeling the dependency among labels" and then applying knowledge distillation with on-the-fly naive ensembling (Fig. 5).[10]
This year's top three approaches can be broken down as follows.

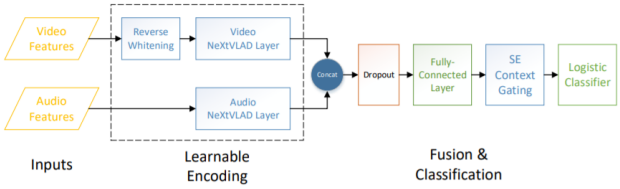


**Figure 5: The solution architecture for 2018's YT8M 3rd place winner, taken from its corresponding conference paper.[10]. "FC" denotes a fully connected layer.**

(1) Layer6 AI created a candidate generation pipeline. Videos, i.e. sequences of frame-level features, were used for video-level candidate generation to make the search space of segments smaller. Only videos that are likely to have segments with a certain class were used, whose label is binary (present or not). Class probabilities were calculated for videos using target class $c$ to sort videos by the probability that $c$ will occur and take the top $k$ candidate videos as input for the segment model. The segment-level model classifies the videos. The pairwise architecture takes a segment and target class and outputs the probability they're related. The result is a single pairwise model of segment-class "relevance" associations

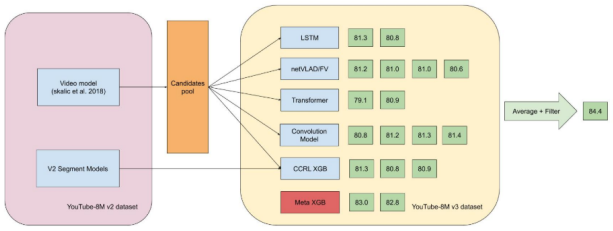


**Figure 6: The solution architecture for 2019's YT8M 1st place winner, taken from its Kaggle leaderboard forum post.[1]**

modeled (Fig. 6).[12]

(2) BigVid Lab from Fudan University created an ensemble of feature aggregation. The team, too, applied feature quantization methods and used Gated Deep Bag-of-Frames, Soft Deep Bag-of-Frames, NetVLAD, WetFV, and RestNetLike approaches. Their team chose a weighted binary cross-entropy loss "to increase the influence of positive samples" and used video-level predictions to "filter out false-positive segment predictions" (Fig. 7). [22] For their pipeline, they first pretrain the base models at a video level but use segment-level data to fine-tune. Sequence features are extracted at the frame level but are aggregated (Fig. 8).[22] They use a mixture architecture with KL divergence to get an "extra regularization term" and also apply knowledge distillation. For segment-level fine-tuning, they use "weighted cross-entropy loss". Lastly, there's an inference strategy of creating 1000 minimum heaps where segment predictions are pushed (Fig. 9). Predicted labels are removed if there are no candidate classes, so false positive examples are eliminated. The final ensemble is used as a single model based on stochastic weight averaging for robustness, though the authors note that there is not much improvement.[22]

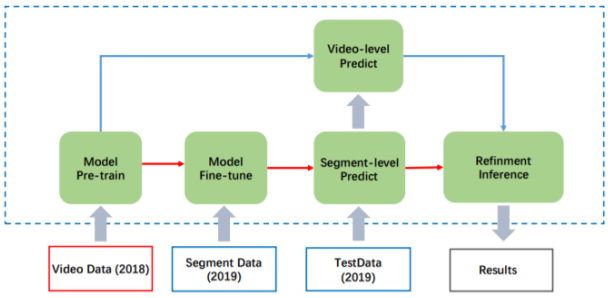


**Figure 7: The solution architecture for 2019's YT8M 2nd place winner, taken from its corresponding conference paper.[22]**

(3) The RLin team from the University of North Carolina at Charlotte created a deep mixture model using online knowledge distillation (MOD). There was a 2-layer online distillation architecture spanning four MixNeXtVLAD models, each of which contains three NeXtVLAD models.[11] They started by also using candidate generations with video-level classifiers and took the top twenty topics for every video to "reduce the search space".[11] Then the segment-level classifier assigned probabilities to the 5-second segments in each video for fine tuning.

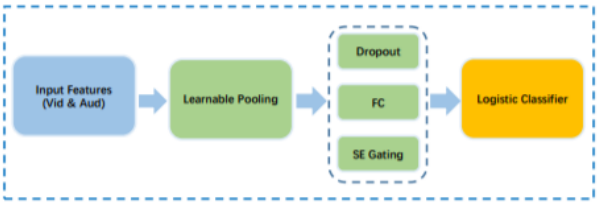


**Figure 8: The frame-level architecture for BigVid Lab's 2nd place approach, taken from its corresponding conference paper.[22] "FC" denotes a fully connected layer.**

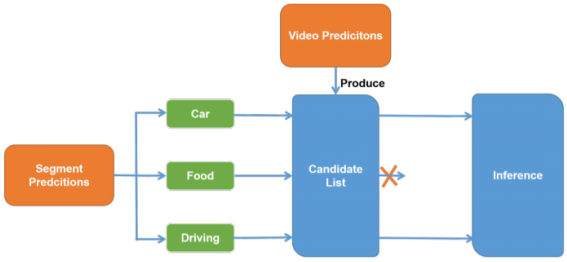


**Figure 9: The solution architecture for BigVid Lab's inference strategy. Taken from its corresponding conference paper.[22]**

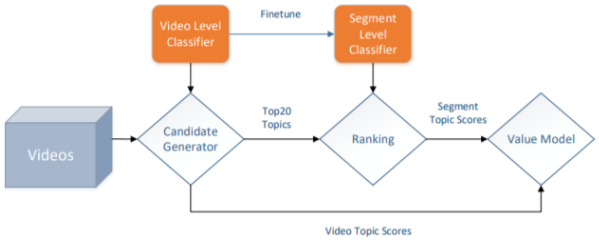


**Figure 10: The solution architecture for RLin's 3rd place approach. Taken from its corresponding conference paper.[11]**

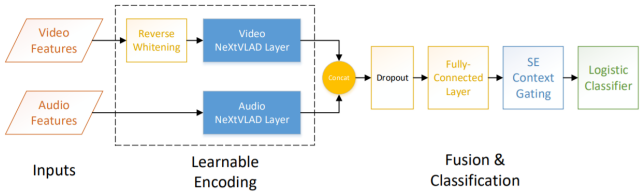


**Figure 11: The frame-level solution architecture for Rlin's approach. Taken from its corresponding conference paper.[11]**

# 3 ANALYSIS

## 3.1 Comparisons, Performance, Trends

Two baseline methods were applied by the creators of the YT8M dataset, namely Deep Bag-of-Frames (DBoF) and LSTM. In the former, they applied RELU activation layers with max pooling for feature aggregation, and trained using stochastic gradient descent with logistic loss and cross-entropy loss. In the LSTM baseline, binary classifiers were trained over all data for each label; layers were not fine-tuned. Methods performed well at a frame level but not video level, which led to the conclusion that "video-level prediction task cannot be reduced to simple frame-level classification".[3] The authors also note that batch optimization classifiers like SVMs are "unfeasible" and require the use of online learning algorithms, which are common to the top three performers of the challenge.[3] [12] [22] [11]

The top three performers in 2019's competition have a fair amount of overlap in methodology. Two of the three use candidate generation,[12][11] two of the three use mixture models,[22][11] and all three first train on frame-/video-level features before moving to segment-level training or tuning and use knowledge distillation in some form to compress their models. Though ensembles performed best, a number of base models across the experiments achieved relatively high MAP values (Fig. 1).

The chief architectural differences come from Layer 6 AI's approach, which leaves out SE context gating. All three use slightly different inference strategies, but the main pipeline is very similar across the board (Fig. 6, 8, 11).

One limitation of using the performance benchmark from YT-8M is that other metrics that might yield valuable insights. Firstly, I want to know why the competition moved away from using the GAP metric and instead to using the MAP metric other than for its popularity, which was vaguely implied in some of the literature but not discussed at length. A confusion matrix, for instance, would be helpful in assessing the relative values of false positives in misclassifications. This would not necessarily always be relevant, but could have varying impacts depending on the domain. False positives in the criminal justice system, or false negatives (in particular) for medical diagnoses, for instance, would reflect this. In addition, it would provide insight into how the model is skewed - and therefore, how relative weights may improve the performance. In Kaggle's private leaderboard, which calculated the MAP using 80% of the test data, the top seven teams all had MAP scores between 80% and 84% (Table 2).Though this discrepancy is worth noting, other measures would help further compare between methods - including final model size and speed. If the improvements between implementations are not substantial, it would be interesting to see how these algorithms generalize to new data, such as "unpopular" videos or edge cases that had previously been filtered out. The scalability would also be useful to test.

## 3.2 Reflection: "What would I have done?"

My approach would also aim for a two-stream architecture. Originally my plan was to use a 2-dimensional CNN for feature extraction,[4] and then apply a 1-dimensional CNN on the sequence of frames using batch renormalization and $K$-fold cross-validation. Similarly to the methods above, I would have trained first on the frame-level data (2-D CNN) and then trained on segment-level data (1-D CNN). In retrospect, a brute-force CNN would probably achieve very suboptimal performance. However, CNNs generally perform faster than RNNs, so i would have tried it to compare the tradeoff between speed and accuracy. I would have also tried different online learning methods with various initializations - not just stochastic gradient descent - to fine-tune the model and to see which methods performed better.

# 4 DISCUSSION

There are various extensions for this line of research, and given the previous success of the challenge, I think it is likely that Google Research will continue to host Kaggle competitions for the YT-8M dataset and its variants. Because model constraints were added to the challenge from year 1 for year 2,[18][19] I think it is likely that temporal localization will be subject to similar budget constraints. Rather than restricting only the model size, there might be interest in reducing the runtime costs further. Because there are many application domains for computer vision as a whole, I think future challenges may depend on the active sociopolitical climate. The rise of companies like Clearview AI and other controversial endeavors may call for security applications,[8] like obfuscation and blurring for data privacy reasons. They may also further refine the temporal granularity - segments were divided into 5-second chunks, but perhaps they will want to pinpoint it to the nearest microsecond. Another possible extension would be audiovisual synchronization namely in mapping audio features to visual features. This could be further extended to deepfake detection; for example, temporally localizing audio features like distortions or unnatural reverberation, or checking for a jumpcut that would create analogous visual anomalies.

There are extensions I would have liked to explore given more time. For the literature review, I would have liked to build and test the models on the same hardware. The first reason is to test for repeatability. The second is to get a fairer, more accurate benchmark comparison. I would have also liked to gather more metrics beyond mAP scores (compared on Kaggle), such as precision and recall. It would have also been nice to generate visualizations like ROC curves or heat maps to compare the accuracies. I took the metrics

**Table 1: Base Model Performances, compiled from the conference papers.**

| Team | Base Model | MAP Score |
|---|---|---|
| Layer6 AI | CNN | 0.8036 |
| Layer6 AI | LSTM | 0.8023 |
| Layer6 AI | Transformer | 0.7955 |
| Layer6 AI | NetVLAD | 0.8023 |
| Layer6 AI | NetFV | 0.8028 |
| Layer6 AI | CCRL | 0.8091 |
| BigVid Lab | Mix-NeXtVLAD | 0.81548 |
| BigVid Lab | Mix-EarlyNetVLAD | 0.81212 |
| BigVid Lab | Mix-LightNetVLAD | 0.8093 |
| BigVid Lab | Mix-GatedDBOF | 0.81327 |
| BigVid Lab | Mix-SoftDBOF | 0.81421 |
| BigVid Lab | Mix-NetFV | 0.81421 |
| BigVid Lab | Mix-GRU | 0.8058 |
| BigVid Lab | Mix-ResNetLike | 0.80928 |
| BigVid Lab | Mix-ResNetLike-Imbalance | 0.81034 |
| BigVid Lab | Mix-ResNetLike-Concat | 0.81100 |
| RLin | NeXtVLAD | 0.79642 |
| RLin | NeXtVLAD_large | 0.80586 |
| RLin | NeXtVLAD_distill | 0.81509 |

**Table 2: Final 2019 Leaderboard Results**

| Team | MAP Score |
|---|---|
| Layer6 AI | 0.83292 |
| BigVid Lab | 0.82620 |
| RLin | 0.82551 |
| bestfitting | 0.81707 |
| Last Top GB Model | 0.80459 |
| ByteVideo | 0.80363 |
| Ceshine | 0.80099 |

they output in their papers, but beyond that, I would have liked to add more visuals. Given more time and computational resources, I would also implement my approach and apply it on the same hardware to compare it with the other methods.

## 5 CONCLUSION

In this paper, I summarized and analyzed some of the prevailing methods in computer vision research applied to the YouTube-8M Video Understanding Challenge. Many of them used candidate generation, neural networks, and VLAD model representation in a two-stream architecture. Next, I discussed a basic approach to the problem. Lastly, I reviewed potential extensions and implications of the challenge.

## ACKNOWLEDGMENTS

Here I wish to acknowledge some of the people and programs in my journey. Firstly, I would like to thank my thesis advisor, Joe Johnson, for not only his dedication to the engineering program, but also for the support and advice this year. There were several hiccups along the way, and I appreciate the time and the help he has provided. I would also like to thank my honors advisor, Zhijie Jerry Shi, for his role in making my senior year progress more smoothly and helping me manage my honors requirements. I would like to thank the staff members at the High Performance Computing Facility for their support (and amazingly fast responses) in setting me up with an account and troubleshooting data transfer issues and other user concerns. I would like to thank the Computer Science Engineering department for all of the courses, programming, events, and resources from which I have been able to benefit since I began my studies at the University of Connecticut. I would like to thank the Honors Program and the STEM Scholar Community for the network of bright and creative minds, as well as the resources and opportunities to pursue independent projects. And finally, I would like to thank my family and friends for their enduring support of my academic career and well-being. I would not be where I am today and am very grateful for all of the support I have been given up until this point. Thank you.

# REFERENCES


[1] [n.d.]. 1st Place Solution, URL=https://www.kaggle.com/c/youtube8m-2019/discussion/112869, author=Jeremy Ma.

[2] 2020. Annual Senior Design Demonstration Day. UCONN School of Engineering. https://seniordesign.engr.uconn.edu/wp-content/uploads/sites/160/2020/03/Senior-Design-2020.pdf

[3] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Apostol (Paul) Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. 2016. YouTube-8M: A Large-Scale Video Classification Benchmark. In *arXiv:1609.08675*. https://arxiv.org/pdf/1609.08675v1.pdf

[4] François Chollet. 2018. *Deep Learning with Python*. Manning, New York.

[5] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[6] Milos Hauskrecht. 2004. Ensamble methods.Mixtures of experts. https://people.cs.pitt.edu/~milos/courses/cs2750-Spring04/lectures/class22.pdf

[7] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. 2017. CNN Architectures for Large-Scale Audio Classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. https://arxiv.org/abs/1609.09430

[8] Kashmir Hill. 2020. The Secretive Company That Might End Privacy as We Know It. (2020). https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html

[9] Stanford Vision Lab. [n.d.]. About ImageNet. http://image-net.org/about-overview

[10] Rongcheng Lin, Jing Xiao, and Jianping Fan. 2018. NeXtVLAD: An Efficient Neural Network to Aggregate Frame-level Features for Large-scale Video Classification. https://static.googleusercontent.com/media/research.google.com/en//youtube8m/workshop2018/c_03.pdf

[11] Rongcheng Lin, Jing Xiao, and Jianping Fan. 2019. MOD: A Deep Mixture Model with Online Knowledge Distillation for Large Scale Video Temporal Concept Localization. https://static.googleusercontent.com/media/research.google.com/en//youtube8m/workshop2019/c_05.pdf

[12] Junwei Ma, Satya Krishna Gorti, Maksims Volkovs, and Ilya Stanevich. 2019. Cross-Class Relevance Learning for Temporal Concept Localization. https://static.googleusercontent.com/media/research.google.com/en//youtube8m/workshop2019/c_02.pdf

[13] Antoine Miech, Ivan Laptev, and Josef Sivic. 2017. Learnable pooling with Context Gating for video classification. https://static.googleusercontent.com/media/research.google.com/en//youtube8m/workshop2017/c11.pdf

[14] Alexander Rakhlin. [n.d.]. Online Methods in Machine Learning. https://www.mit.edu/~rakhlin/6.883/lectures/lecture01.pdf

[15] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. *CoRR* abs/1406.2199 (2014). arXiv:1406.2199 http://arxiv.org/abs/1406.2199

[16] Miha Skalic and David Austin. 2018. Building a Size Constrained Predictive Model for Video Classification. https://static.googleusercontent.com/media/research.google.com/en//youtube8m/workshop2018/c_14.pdf

[17] Yusuke Uchida. 2016. Local Feature Detectors, Descriptors, and Image Representations: A Survey. *CoRR* abs/1607.08368 (2016). arXiv:1607.08368 http://arxiv.org/abs/1607.08368

[18] Google Research Video Understanding. 2017. The 3rd YouTube-8M Video Understanding ChallengeCan you produce the best video tag predictions? https://www.kaggle.com/c/youtube8m/overview

[19] Google Research Video Understanding. 2018. The 2nd YouTube-8M Video Understanding ChallengeCan you create a constrained-size model to predict video labels? https://www.kaggle.com/c/youtube8m-2018

[20] Google Research Video Understanding. 2019. The 3rd YouTube-8M Video Understanding ChallengeTemporal localization of topics within video. https://www.kaggle.com/c/youtube8m-2019

[21] Google Research Video Understanding. 2019. YouTube-8M Segments Dataset. https://research.google.com/youtube8m/index.html

[22] Jejia Weng, Rui Wang, and Yu-Gang Jiang. 2019. Exploring the Consistency of Segment-level and Video-level Predictions for Improved Temporal Concept Localization in Videos. https://static.googleusercontent.com/media/research.google.com/en//youtube8m/workshop2019/c_13.pdf